MAG Models for the Marginal Distributions of DAG Models

Candidate Number: 104042

Abstract

Directed acyclic graph (DAG) models are used to define classes of distributions which satisfy conditional independence constraints implied by the associated graphs. If we consider the marginal distribution of a distribution which obeys a DAG model, the conditional independence constraints which are implied cannot generally be represented using a DAG. There have been several classes of mixed graphical models which have been introduced that deal with this problem.

In this paper we focus on the class of mixed graphs known as maximal ancestral graphs (MAGs). We show how a MAG can be constructed so that the conditional independence constraints implied by the margin of a DAG hold. We then consider the relative merit of using MAGs as opposed to DAGs in graphical models to represent both the margins of DAGs, and also other data sets with hidden variables.

1 Introduction

We initially consider the problem of representing the conditional independences of a joint distribution over a set of random variables. Specifically, the set of independences defined by the joint distribution can often be represented visually using a graph, in which vertices represent the variables, and the edges represent some form of dependence between the variables. This is useful in terms of being able to interpret the nature of the relationships between variables. The definition of a Markov property allows a formal way of determining which conditional independences hold for a given graph. The combination of the property and the graph defines a graphical model.

There are two classes of graphs which were commonly used initially [1]. Undirected graphs (UGs), which allow only one edge type with no arrows, and directed acyclic graphs (DAGs), which allow only edges with one arrowhead. The DAGs provide a particularly intuitive interpretation, in that an arrow pointing from one vertex to a second one implies that the distribution of the second is somehow dependent on the first.

This paper is concerned with the properties of DAG graphs, particularly the fact that the graphical models resulting from them are not closed under marginalisation, that is, the margin of a joint distribution which obeys a DAG model can not be represented accurately by a DAG model [2]. This invites the study of the resulting marginal distributions, and whether they can be represented faithfully using a graphical model.

There have been many classes of graph which have been used to address this problem. This paper will focus on one particular type of graph, a maximal ancestral graph (MAG) introduced by Richardson and Spirtes[3], which is a graph with three types of edge. We will be investigating how well the graphical models resulting from this richer class of graphs can represent the marginal distribution of a DAG model, with a focus on the conditional independence constraints implied by the marginal distribution. We will also be covering the usefulness of the larger class of MAG models, in comparison with the class of DAG models, when trying to learn the underlying distribution of a sample data set. An example of this kind of graph is shown in Figure 1.

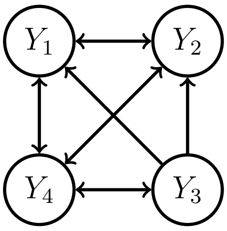

Figure 1: A MAG

One difficult task when using graphical models to model data is learning the structure of the graph to be used in the model. In this paper we will be using a score-based method to find the best graph to use, in particular the Bayesian information criterion (BIC) which is a likelihood score which penalises large numbers of parameters. As the number of MAGs scales exponentially with the number of variables, it is not feasible to compute the BIC score for each MAG model and find the global optimum [4]. Instead we will run a greedy search algorithm using the BIC score to find the best local optimum model, which will then be our final fitted graphical model.

1.1 Outline of the paper

In section 2 we give an introduction into graphical models, including directed and undirected graphs, and also the marginalisation problems of DAGs. In section 3 we outline some of the various classes of graphical model which have been used to solve these issues. In section 4 we introduce the algorithm that is used to find the optimum graph to represent independences, and then present empirical results for the various simulated data sets described above.

2 Undirected graphs, directed acyclic graphs and the marginalisation problems of directed acyclic graph models

This section will outline the basic graph and probability theory required for the models under consideration, then briefly summarise Lauritzen's [1] introduction to undirected graphical models and directed acyclic graphical models and finally discuss the fact that the class of DAG models is not closed under marginalization.

2.1 Basic graph theoretical concepts

A *graph* is a pair of sets $G = \{V, E\}$ where V is the set of *vertices* and E is the set of *edges*. Edges can take various forms including *undirected* $-$, *directed* $\rightarrow$ and *bi-directed* $\leftrightarrow$. There are other classes of graph which have dashed lines of various types, but they are not covered in this paper. Figure 2 is an example of a graph with undirected edges and Figure 4 is an example with only directed edges.

If there is an edge between i and j in a graph G, then we say that the two vertices are *adjacent* in G. For an undirected graph we write $i \sim j$ if $\{i, j\} \in E$. The set of vertices which are adjacent to i is called the *boundary* of i and is denoted $bd_G(i)$. For example, in Figure 2 $bd_G(5) = \{1, 2, 4\}$. A *path* is a sequence of adjacent vertices without repetition. In Figure 2, $1 - 2 - 3$ is a path.

For a directed graph the set of vertices which have an edge pointing to i is the set of *parents* of i and is denoted $pa_G(i)$. If there is a directed path from i to j then j is a *descendant* of i. We also say that i is a descendant of itself. We refer to $nd_G(i)$, the set of vertices which are not descendants of i. The *ancestors* of i are the vertices which have a directed path to i. We can use the parents and non-descendants of sets as well, using a slight but convenient abuse of notation. In Figure 4, the $pa_G(2) = \{1, 3\}$ and $nd_G(2) = \{1, 3, 4, 5\}$. Where it is clear which graph we are referring to, we may leave out the G subscript.

For an undirected graph, we say that a set of vertices A is *separated* from another set B by a set C if every path from $a \in A$ to $b \in B$ has at least one vertex in C. We write this as $A \perp_s B \mid C[G]$. In Figure 2, for example, $1 \perp_s 4 \mid 2, 5$.

For a graph $G = \{V, E\}$ and $W \subseteq V$ we define the *induced subgraph* G_W as the set of vertices W and the edges in E such that both endpoints are in W. A subgraph is *complete* if every pair of vertices in the subgraph has an edge between them. A maximal complete subgraph of a graph G is a *clique* of G.

2.2 Conditional Independence

Given a probability distribution p on a set of variables, we write that two variables A and B defined on a product space $\mathcal{A} \times \mathcal{B}$ are *conditionally independent* given a variable C on the space $\mathcal{C}$ if:

$$p(a, b \mid c) = p(a \mid c)p(b \mid c) \text{ for all } a \in \mathcal{A}, b \in \mathcal{B} \text{ and } c \in \mathcal{C}$$

This is written as $A \perp\!\!\!\perp B \mid C[p]$. This definition can be straightforwardly extended to disjoint sets $A, B \text{ and } C$ of random variables.

2.3 Undirected Graphs

2.3.1 Markov Properties of Undirected Graphical Models

An *undirected graph* is a graph which contains only undirected edges. In order to define a graphical model on a graph, we need to consider each vertex v as having a random variable X_v associated with it. The graphical model associated with a graph is defined via the joint distribution of the associated variables constrained by the conditional independences implied by the missing edges of the graph. We use the following Markov properties to define how the missing edges translate into conditional independences for undirected graphs.

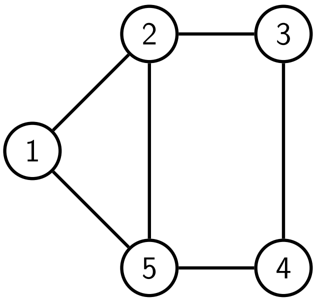

Figure 2: An undirected graph

We say that a distribution p on the variables X_V associated with a graph $G = \{V, E\}$ satisfies the *pairwise Markov property for G* if:

$$i \nsim j \Rightarrow X_i \perp\!\!\!\perp X_j \mid X_{V \setminus \{i,j\}}[p].$$

This property indicates that when an edge is missing from the graph G, there is a corresponding conditional independence between the two endpoints given all the other vertices. For example, in Figure 2 this property has implications including that $X_1 \perp\!\!\!\perp X_3 \mid X_2, X_4, X_5$ and $X_2 \perp\!\!\!\perp X_4 \mid X_1, X_3, X_5$.

A probability distribution p satisfies the *local Markov property for G* if for all $i \in V$ we have:

$$X_i \perp\!\!\!\perp X_{V \setminus (bd_G\{i\}, \{i\})} \mid X_{bd_G\{i\}}[p].$$

This property states that the variable X_i is conditionally independent of all the other variables given the variables corresponding to the adjacent vertices. In Figure 2, the local Markov property implies, for example, $X_5 \perp\!\!\!\perp X_3 \mid X_1, X_2, X_4$ and $X_3 \perp\!\!\!\perp X_1, X_5 \mid X_2, X_4$.

A distribution p satisfies the *global Markov property for G* if for disjoint sets $A, B \text{ and } C$ we have that:

$$A \perp_s B \mid C[G] \implies X_A \perp\!\!\!\perp X_B \mid X_C[p].$$

This property is the most general of the three stated Markov properties, and uses the concept of separation of vertices to define conditional independences from a graph. This property implies $X_1 \perp\!\!\!\perp X_4 \mid X_2, X_5$ in Figure 2, which cannot be directly obtained from the other two Markov properties.

Finally, it is useful to define the factorization criterion for a distribution p, and we say that p *factorizes according to* G if, for the set of cliques of G, $\mathcal{C}(G)$, we have:

$$p(x_V) = \prod_{C \in \mathcal{C}(G)} \varphi_C(x_C).$$

We call φ_C the potential of the clique.

By the definitions of the three Markov properties we have that the global Markov property implies the local Markov property which in turn implies the pairwise Markov property. Hence the following two theorems complete the equivalence between the three Markov properties and the factorization criterion for a distribution p, provided that p is strictly positive:

Theorem 2.1. (Factorization implies the global Markov property)

For any undirected graph G and any probability distribution p on $\mathcal{X}$ it holds that if p satisfies the factorization criterion for G, then it satisfies the global Markov Property for G.

Theorem 2.2. (Hammersley and Clifford)

If a probability distribution p such that $p(x_V) > 0$ satisfies the pairwise Markov property for a graph G, then p factorizes according to G.

The proofs can be found in [1] and are omitted here. Following the equivalence of the properties above, we call a positive distribution that satisfies them *Markov* for G.

The aim of this paper is to investigate the issue of closure under marginalization for directed acyclic graphical models. We first define what closure under marginalisation means. We say that a class of graphical models is *closed under marginalisation* if when we marginalise a distribution which obeys some model in that class, the conditional independence constraints which the marginal distribution implies can be described exactly, without any additional independence constraints using another graphical model in that class.

For undirected graphs however, this is not an issue as we can use a *rubber band procedure*[5] which is summarised as follows.

Algorithm 2.1. *(marginalizing undirected graphs over a vertex set K)*
For every vertex $k \in K$ and every pair i, j such that $i \sim k$ and $j \sim k$ add an edge $i \sim j$ and remove the vertex k and all the edges with an endpoint at k.

For example, in Figure 2 if we marginalize over the variable X_5, we get the resulting graph in Figure 3. The distributions which obey the graphical model associated with this graph also obey all the conditional independence constraints of the original graphical model on the four variables which do not contain X_5.

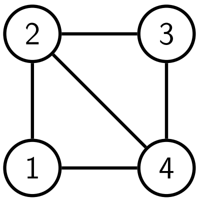

Figure 3: the undirected graph after marginalizing over X_5 in Figure 2

2.4 Directed Acyclic Graphs

2.4.1 Markov Properties of Directed Acyclic Graphical Models

A *directed acyclic graph (DAG)* is a graph that contains only directed edges, in addition to not containing any directed cycles. An example is shown in Figure 4. The importance of the class of statistical models associated with these graphs is clear, as we can often presume that random variables affect one another in a non-symmetrical fashion. We can define the following Markov properties for DAG models.

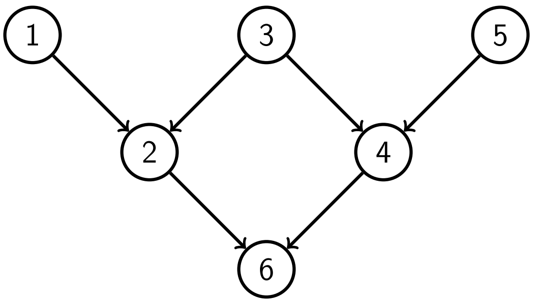

Figure 4: A DAG $\mathcal{G}$

Given a directed acyclic graph $\mathcal{G}$ with vertex set V, we say that the distribution $p(x_V)$ *factorizes with respect to $\mathcal{G}$* if

$$p(x_V) = \prod_{v \in V} p(x_v \mid x_{pa_G\{v\}})$$

for all $x_V \in \mathcal{X}_V$. This gives us a clear conditional independence model for $\mathcal{G}$, as for any ordering of the vertices in V, $\{x_1, x_2, ..., x_k\}$ we can form the following factorization of a distribution p

$$p(x_V) = \prod_{i=1}^{k} p(x_i \mid x_1, ..., x_{k-1})$$

We say that a distribution p satisfies the *local Markov property* for a DAG $\mathcal{G}$ if for all $v \in V$,

$$X_v \perp\!\!\!\perp X_{nd_{\mathcal{G}}(v) \setminus pa_{\mathcal{G}}(v)} \mid X_{pa_{\mathcal{G}}(v)}[p]$$

For example, for a distribution that is *Markov* for $\mathcal{G}$ in Figure 4, the local Markov property implies that

$$\begin{aligned} X_1 \perp\!\!\!\perp X_3, X_4, X_5 & \qquad & X_2 \perp\!\!\!\perp X_4, X_5 \mid X_1, X_3 \\ X_3 \perp\!\!\!\perp X_1, X_5 & \qquad & X_4 \perp\!\!\!\perp X_1, X_2 \mid X_3, X_5 \\ X_5 \perp\!\!\!\perp X_1, X_2, X_3 & \qquad & X_6 \perp\!\!\!\perp X_1, X_3, X_5 \mid X_2, X_4 \end{aligned}$$

It is also useful to define a global Markov property for DAGs aswell. A set of vertices $A \subseteq V$ is *ancestral* if it contains all of its own ancestors. We define a *v-structure* in a graph as where three vertices i, j, k are such that $i \to k$ and $j \to k$. We define the *moral graph* of a DAG $\mathcal{G}$ as an undirected graph on the vertices V of G, such that there is an edge between i and j if there is an edge between the two in $\mathcal{G}$ or if i and j form part of a v-structure in $\mathcal{G}$.

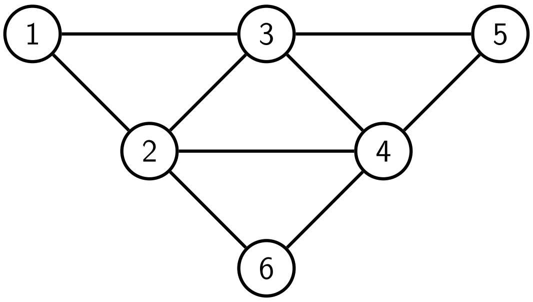

Figure 5: The moral graph of $\mathcal{G}$, $\mathcal{G}^m$

We say that $p(x_V)$ satisfies the *global Markov property* with respect to $\mathcal{G}$ if when A and B are separated by C in $(\mathcal{G}_{an(A\cup B\cup C)})^m$ then $X_A \perp\!\!\!\perp X_B \mid X_C[p]$.

For the graph $\mathcal{G}$ in Figure 4, the moral graph in Figure 5 shows that the global Markov property gives the several further conditional independencies, including $X_6, X_2 \perp\!\!\!\perp X_5 \mid X_3, X_4$. It also shows us that whilst there is no edge between vertices 1 and 3, it does not hold that $X_1 \perp\!\!\!\perp X_4 \mid X_2$ as in the moral graph there is an edge between 1 and 3.

2.4.2 Markov Equivalence

It is important to note that different graphs can have the apply the same conditional independence constraints. Such graphs are called Markov equivalent. It is easy to see that this forms an equivalence relation. For example, the two graphs in Figure 6 are in the same equivalence class, as they have the same v-structures and skeleton, and so the same moral graph, but they clearly have different implications in terms of causality. We would require extra information (for example, expert knowledge) in order to correctly orientate the edges in a graphical model with directed edges.

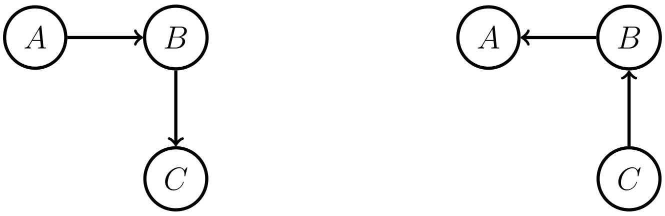

Figure 6: Two Markov equivalent DAGs, with different causal implications.

We note that changing the orientation of the edges can have an effect on the conditional independences implied by the graphs, so changing the equivalence class of a graph, as in Figure 7.

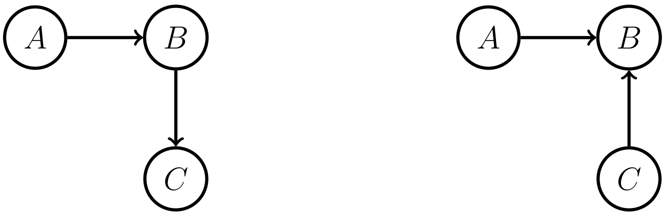

Figure 7: Two DAGs which have the same skeleton, but are not Markov equivalent. The moral graph of the second graph would have an edge between A and C.

2.4.3 An Example of a Fitted DAG Model

We will now present an example of fitting a DAG model to a data set, to make it clear what these objects look like in practise. To do this, we take a sample over four variables, of a 100 observations, which has the following sample covariance matrix. We can do this because the sample covariance is a sufficient statistic for a normal distribution with known mean vector, in this case the zero vector.

$$\begin{bmatrix} 84.5 & 11.4 & 104.2 & 113.2 \\ 11.4 & 94.4 & 99.6 & 106.8 \\ 104.2 & 99.6 & 305.6 & 334.9 \\ 113.2 & 106.8 & 334.9 & 470.3 \end{bmatrix}$$

We will call the variables X_1, X_2, X_3, X_4. Before we can fit parameters and find the fitted model, we need to decide on a DAG to define the graphical model we will be fitting. We start using the graph found in Figure 8.

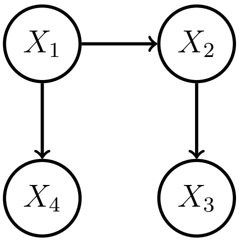

Figure 8: The first selected DAG

Throughout the paper, we will be assuming that the underlying distribution of the data is a multivariate Gaussian distribution. We can use the ggm package for R to fit a DAG model with the given DAG above. We then obtain the following regression equations:

$$X_2 = 0.043X_1$$

$$X_3 = 1.22X_2$$
$$X_4 = 0.94X_1$$

The model has 3 degrees of freedom, and a deviance of 197.44.

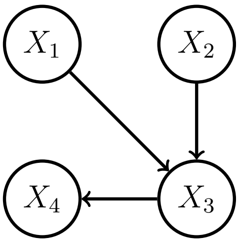

Figure 9: The second selected DAG

If instead, we use the graph found in figure 9, we obtain another graphical model, which when fitted to the data set results in the following regression equations:

$$X_3 = 0.97X_1 + 1.19X_2$$
$$X_4 = 1.02X_3$$

This model also has 3 degrees of freedom, and a deviance of 2.29. As we can see, this model has a significantly lower deviance than the previous one. This leads us to believe the second model fits the data better. We would then choose the graph in Figure 9 over the graph in Figure 8 to represent the conditional independences of the data.

2.4.4 Marginalization Problems with DAG Models

We are now in a position to examine the models of the margins of DAG models. We begin by using the example found in [6]. The DAG in Figure 10 implies the following conditional independences for any distributions obeying the Markov property on that graph:

$$Y_1 \perp\!\!\!\perp Y_3, Y_4, X, \qquad Y_2, X \perp\!\!\!\perp Y_4, \qquad Y_3 \perp\!\!\!\perp Y_2 \mid X.$$

We now consider marginalizing over the variable X, imagining that we are unable to measure the variable, or it is unobserved. We therefore remove the conditional independence constraint $Y_2 \perp\!\!\!\perp Y_3 \mid X$. It is now not possible to represent the marginal distribution on the remaining variables using a DAG. If we try the DAG in Figure 11, we see that this graph implies that $Y_4 \perp\!\!\!\perp Y_2 \mid Y_3$,

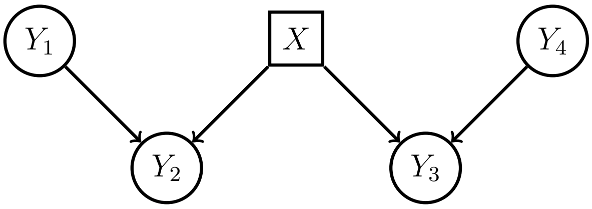

Figure 10: A DAG

which is clearly not the case in Figure 10. If we try to avoid this by adding another edge such as $Y_2 \rightarrow Y_4$ so this constraint does not hold, we can no longer express that $Y_2 \perp\!\!\!\perp Y_4$.

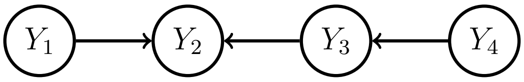

Figure 11: A possible marginal DAG

We see that we are unable to express all of the conditional independence constraints in the original DAG using another DAG on the new set of vertices, excluding X. There is a clear need for another kind of edge, either undirected or bi-directed, in order to represent the relationship between the variables in the marginal model when there are latent variables in the complete model. This is part of the motivation behind several classes of graphical model, which are outlined in the next section.

3 Some Extensions of Directed Acyclic Graphs

There have been many classes of graphical model which have been devised in order to capture varied and more complex interactions between variables, than is possible with just a directed acyclic graph. We will now introduce several of these models and demonstrate their uses and limitations.

3.1 Maximal Ancestral Graphs

The primary class of graphical models we will examine are a subclass of mixed graphs. This class was developed by Richardson and Spirtes [3] specifically because it is closed under marginalization and conditioning.

3.1.1 Ancestral Graphs

We first need some definitions. A *mixed graph* is a graph that contains up to three different types of edges, undirected, bi-directed and directed. A vertex j is *anterior* to a vertex j if there exists a path of undirected and directed edges all orientated towards i. We define the set of vertices anterior to i as $ant(i)$. For a set of vertices X, we define $ant(X) = \{i \mid i \in ant(j) \text{ for some } j \in X\}$. We say that for a mixed graph if $i \leftrightarrow j$ then j is a *spouse* of i. The set of spouses is written as $sp(i)$. Also if $i - j$ then j is a *neighbour* of i and the set of neighbours is denoted $ne(i)$. We say a mixed graph G is an *ancestral* graph if the following conditions hold for all vertices i in G:

1. $i \notin ant(pa(i) \cup sp(i)) \cup pa(i) \cup sp(i)$;
2. if $ne(i) \neq \emptyset$ then $pa(i) \cup sp(i) = \emptyset$.

Note that DAGs are a subclass of ancestral graphs. Figure 12 provides an example of a typical ancestral graph.

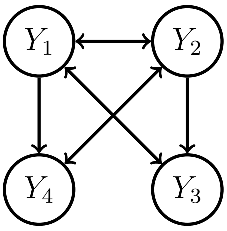

Figure 12: An ancestral graph

As before, we define a separation criterion to allow a global Markov property for ancestral graphs. This is an extension of d-separation found in DAGs. A nonendpoint vertex v on a path is a *collider* on the path if both the edges on the path with v as an endpoint have arrowheads at v. For example, in Figure 12, Y_4 is a collider on the path $Y_1 \rightarrow Y_4 \leftrightarrow Y_2$. Any nonendpoint vertex which is not a collider is a *noncollider* We say that a path between i and j is *m-connecting* given a set Z with $i, j \notin Z$, if:

1. every noncollider on the path is not in Z, and
2. every collider on the path is in $ant(Z) \cup Z$

If there is no path m-connecting i and j given Z, then i and j are *m-separated* given Z. Sets X and Y are said to be m-separated given Z if every pair of vertices $x \in X$ and $y \in Y$ are m-separated given Z. We would then write $X \perp_m Y \mid Z[G]$.

We can now define the global Markov property for ancestral graphs. A distribution p satisfies the global Markov property for an ancestral graph G if for any sets of vertices A, B and C,

$$A \perp_m B \mid C \implies X_A \perp\!\!\!\perp X_B \mid X_C[p].$$

3.1.2 Maximal Ancestral Graphs

When we defined the Markov properties for UGs and DAGs, we saw that they both obey local Markov properties, implying that every missing edge represents a conditional independence. This is not the case for any ancestral graph, for example in Figure 12, there is a missing edge between Y_4 and Y_3, but they are m-connected given any subset of $\{Y_1, Y_2\}$. Richardson and Spirtes [3] provides the following subclass of ancestral graphs which allows us to fix this problem. A *maximal* ancestral graph (MAG), is an ancestral graph where for every pair of non-adjacent vertices i and j there exists a set V such that $i, j \notin V$ and $X_i \perp\!\!\!\perp X_j \mid X_Z$. This now adds the constraint that there is a conditional independence for every missing edge. This is formalised in Theorem 3.1 and is proved in [3].

Theorem 3.1. *If G is an ancestral graph, then there exists a unique maximal ancestral graph $\tilde{G}$ formed by adding $\leftrightarrow$ edges to G such that the Markov property implies the same conditional independences for both graphs.*

We now provide the algorithm for generating the marginal ancestral graph after marginalization on a set M.

Algorithm 3.1. *(Generating an ancestral graph $\tilde{G}$ after marginalization over M) Starting from the vertex set $V \setminus M$, and a Markov distribution p we specify edges as follows:*

If i, j are such that for all $Z \subseteq V \setminus (M \cup \{i, j\})$ such that $X_i \perp\!\!\!\perp X_j \mid Z[p]$ does not hold, then use table 1 to choose an edge to be present in the new graph $\tilde{G}$.

$i \in ant(j) \cup \{j\}$;$j \in ant(i) \cup \{i\}$	generates	$i - j$
$i \notin ant(j) \cup \{j\}$;$j \in ant(i) \cup \{i\}$	generates	$i \leftarrow j$
$i \in ant(j) \cup \{j\}$;$j \notin ant(i) \cup \{i\}$	generates	$i \rightarrow j$
$i \notin ant(j) \cup \{j\}$;$j \notin ant(i) \cup \{i\}$	generates	$i \leftrightarrow j$

Table 1: Types of edge induced by marginalization over M

There is a proof in [3] that this algorithm produces a maximal ancestral graph, and that the new graph is indeed the graphical representation of model after marginalization over M. Finally we can once again return to our example DAG in Figure 10. We can use the algorithm above to produce the new MAG in Figure 13. Using the Markov property for ancestral graphs, we see that all the conditional independence constraints the we would expect from marginalising over X in Figure 10 still hold. For example the only path from Y_2 to Y_4 is $Y_2 \leftrightarrow Y_3 \leftarrow Y_4$. The only collider is Y_3 so given the empty set, Y_2 and Y_4 are m-separated and so are independent by the global Markov property.

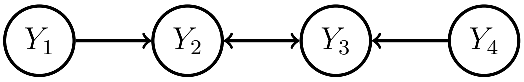

Figure 13: The generated MAG for the marginal graph over X in Figure 10.

3.2 Other Graphs

We have covered MAGs in detail as these are the focus of our simulations later in the paper. Naturally there have been several other classes of graph which have been used to create graphical models which the marginal distribution of a DAG model satisfies. We will outline some of them here. They will not feature in our simulations, however.

3.2.1 Acyclic Directed Mixed Graphs

The class of acyclic directed mixed graphs (ADMGs)[7] are mixed graphs which do not contain any undirected edges, or any directed cycles. For example, the graph in Figure 13 is an ADMG. These graphs not only capture the conditional independence constraints but also additional constraints that the DAG implies such as Verma constraints, see Verma and Pearl (1990). They are not like MAGs, in that they are not simple, so can accept self-loops (edges to themselves).

3.2.2 Marginal Directed Acyclic Graphs

A marginal DAG (mDAG)[2], is a class of graphs which aim to capture not only the conditional independence constraints on the marginal DAG model, but also additional constraints such as Verma constraints, inequality constraints and other constraints (see Richardson and Spirtes [3]). An mDAG can be defined via a DAG and a set of hyper bidirected edges, which represent the marginalised variables. An example is given in Figure 14, which is the mDAG representing the graph of the DAG in Figure 4 after marginalizing over the variable 2, with a hyper-edge between 1,3 and 6.

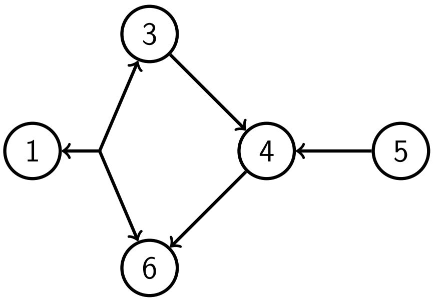

Figure 14: The mDAG representing 4 after marginalization over the node 2.

3.2.3 Chain Graphs

Chain graphs are a well-studied class of mixed graphs. They are defined as graphs containing directed and undirected edges, with no semi-directed cycles (that is, cycles which have either undirected edges or directed edges which are all orientated in one direction). They were introduced by Lauritzen and Wermuth [8] as a natural extension to both UGs and DAGs. Figure 15 provides an example of this class of graphs. There are multiple Markov properties for chain graphs including the multivariate regresson Markov property [9], AMP Markov property [10] and the LWF Markov property [8]. These graphs are not necessarily closed under marginalisation, but there exists an extension for graphs with the LWF Markov property called chain mixed graphs (CMGs) [11] which are closed.

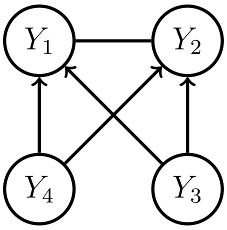

Figure 15: A Chain Graph

3.2.4 Summary Graphs

Summary graphs were defined by Wermuth, Cox and Pearl [12] as an extension of regression graphs [9], which are a subclass of chain graphs in which each edge respects a recursive order of the joint responses [13]. Satisfying distributions have some desirable properties [14]. However, regression graphs as a subclass of chain graphs are not closed under marginalisation, but the extension to summary graphs gives a class of mixed graphs which are. They are defined as follows:

Definition 3.1. Summary Graphs

A graph is a Summary graph if the vertex set V can be partitioned into disjoint sets u and v such that within u the graph has a mixture of a DAG and a covariance graph (which has only dashed edges), and within v it has a UG. Edges between u and v are directed edges pointing at u.

For example Figure 16 shows a summary graph with vertex sets $u = \{1, 2, 3, 4\}$ and $v = \{5, 6, 7, 8\}$. Note that DAGs are just a subclass of summary graphs, and that an ADMG is a summary graph if you change the bi-directed edges to dashed edges.

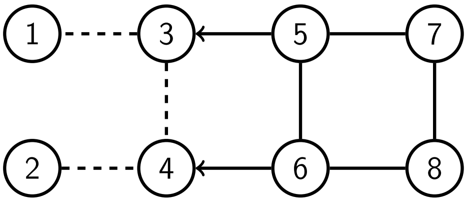

Figure 16: A summary graph

4 Fitting marginal DAG and MAG models

Now that we have outlined some of the possible classes of graphical model that deal with marginalization better than DAG models can, it is useful to see this in action, over a variety of data structures. For this project, DAG and MAG models will be fitted to a variety of marginal data, and we will compare the fitted models with those that we would expect from the previous sections. Firstly, a method is required to find the structure of the DAG or MAG model before we can compare the actual and expected fits.

4.1 Greedy Algorithm for Finding Structure of Models

We will only consider one method for learning the structure of the MAG or DAG which best represents the sample data, a *greedy search algorithm* which is outlined in a paper by Chickering [15] for DAG models. The idea is to repeatedly start at a random model, then find the local optimum using a scoring measure and finally select the best overall model. We will be using the Bayesian information criterion as our scoring measure.

Definition 4.1. Bayesian Information Criterion (BIC)

For a model with maximised likelihood $\hat{L}$, n data points and k estimated parameters, the *Bayesian information criterion (BIC)* is defined as follows:

$$BIC = \log(n)k - 2\log(\hat{L})$$

When we fit a graphical model we will use the degrees of freedom instead of the number of parameters fitted, so up to a constant we have that:

$$BIC = \text{deviance of fitted model} - \log(n) * (\text{degrees of freedom})$$

This is allowed because we are just comparing BIC scores, the number itself doesn't have any particular significance. The BIC is an asymptotic criterion derived under the assumption the underlying distribution of the sample is an exponential family, which is true as all the sample data sets are generated from Gaussian graphical models.

The formal algorithm is provided below:

Algorithm 4.1. *(Greedy Algorithm using BIC for model selection)*

1. *Choose a random starting graph of the relevant class and calculate the BIC of the associated fitted model, G.*

2. *Find the single edge change which gives the valid ancestral graph with associated fitted model G' with the lowest BIC score. If the new score is lower than for the original model, set $G = G'$.*

3. *Repeat step 2. until the BIC score of G' is higher than that of G. Then G is the optimal local model.*

4. *repeat steps 1-3. and choose the local optimum with the lowest BIC score overall. This will be the final model from the algorithm.*

For each model class there are some important points to note. As these steps are not as straightforward as they may initially seem.

Firstly, we have to decide how to choose a random starting structure for the algorithm. For all classes of models, the algorithm is not capable of picking a specific model, rather an equivalence class of the model class. If we pick a graph uniformly at random, then we won't be picking an equivalence class uniformly at random. For example, the graphical model which implies the constraint that all variables are mutually independent has just one DAG representation (the empty graph), whilst the graphical model which applies no independence constraints has $p!$ DAG representations over p variables. MAG models face a similar distributional problem with equivalence classes. This is not an issue for model selection however [15], as for a large enough sample size we will start from every equivalence class at least once with probability close to one, so will find the optimal local model structure from each equivalence class, so also the overall optimal model structure based on BIC. The second problem is that as the algorithm only finds the best model up to the equivalence class, it won't necessarily find the casual relationships in the right orientations, as covered in section 2.4.2.

We will now look at three different types of data, and how our fitting algorithms and the resulting DAG and MAG models capture the true relationships between variables. The first is simply data from the margin of a DAG model.

4.2 Margin of a DAG model

Many real data sets can be viewed as a marginal subset of a DAG model. The latent variables have influence on otherwise related variables in many expert systems. For example, medical knowledge can be viewed as a causal system between invading organisms, pathological states, physical disorders and symptoms [16]. Only the symptoms are observed in this system. It makes sense to examine the effectiveness of DAG and MAG models at explaining the relationships between variables.

Our example is an extension of the graph found in Figure 10. The underlying DAG model for our simulations is as shown in Figure 17. We simulated 8 data sets of varying sample sizes using a Wishart distribution for the sample covariance, given a fixed underlying covariance for the model.

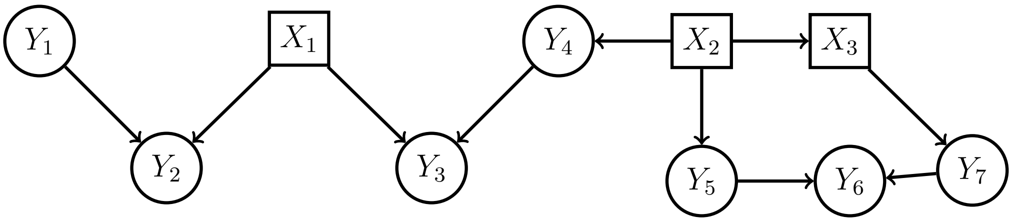

Figure 17: The underlying DAG used in simulation

The variables that we are going to fit DAG and MAG models to are the $Y_i, i \in \{1 : 7\}$, so we can take the sub-matrix of the sample covariance for the underlying data to give us the relevant sample covariance matrix. As we explored in the previous section, it is not possible to represent the independence relationships between these variables using a DAG. However, using a MAG we can represent the marginal conditional independences using the algorithm 3.1. This gives the MAG found in Figure 18.

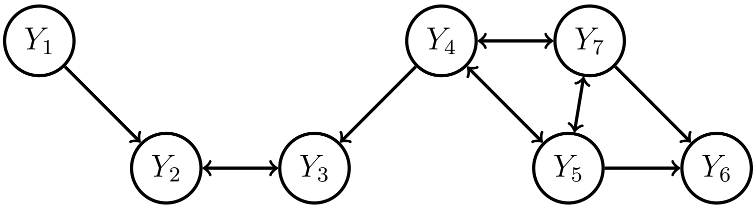

Figure 18: The true marginal MAG for the given variables

The result of this is that we would expect the fitted DAG models to vary in more structure depending on the data, as there is no true underlying DAG whose associated model can fully describe the conditional independences, whereas the fitted MAG models should not only fit the data better as it is a much richer class of models, but it should be more robust to varying data. We will therefore be not only comparing the fitted DAG and MAG models, but also the fitted and true MAG structures.

As expected, learning the structure of the MAG which produces the model with the closest fit to the data takes considerably longer than the DAG equivalent, as there are more edge types to consider at each step of the greedy algorithm. We therefore fitted the MAG model using 100 greedy repetitions and the DAG model using 1000 repetitions. The run-time for the DAG models was around 2.5 hours and it took 3.5 hours to fit the MAG models to the 8 data sets. The output resulted in the the BIC scores found in Figure 19.

For all sample sizes we notice that as expected the learned MAG model does fit the data better than the learned DAG, with a lower BIC score for each sample. The BIC scores cannot be compared across sample sizes however, so the increase in the gap between the two classes of model does not necessarily indicate that the difference in the quality of fit is also increasing.

The chart in Figure 20 gives the BIC score over the course of the greedy algorithm for both the DAG and MAG model for the first data set (the other data sets behaved in a similar manner). Each line corresponds to one repetition of the greedy search, each point representing the BIC score at each step. The behaviour between the two graph types is very similar, with the only noticeable difference being that the MAG runs are less likely to fit the data very well at the beginning of each run, which is expected as there are more possible starting graphs, and less runs in total.

In terms of the structure of the graph produced by the greedy algorithm for DAG graphs, the 8 data sets produced 3 different equivalence classes of graph, and no two identical graphs (as expected because there is no mechanism for

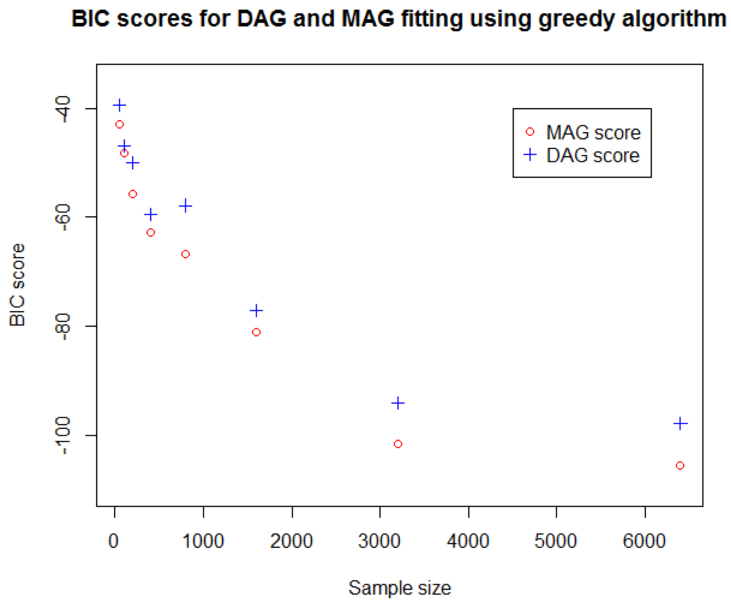

Figure 19: The BIC score for the fitted DAG and MAG for each simulated data set

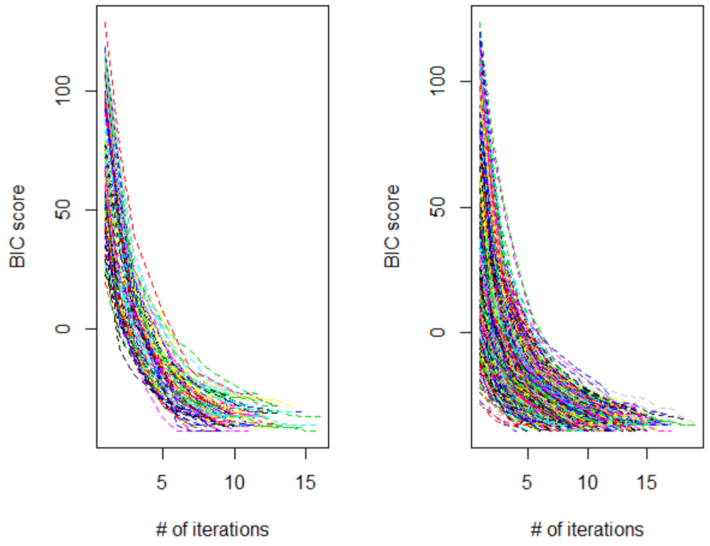

Figure 20: The BIC score at each iteration for the first data set. The left hand graph is for the MAG models, the right hand side for the learned DAG models

choosing edge direction within an equivalence class). An example output for the first data set is shown in Figure 21. This example output once again shows the fact that a DAG cannot express the conditional independences which actually hold.

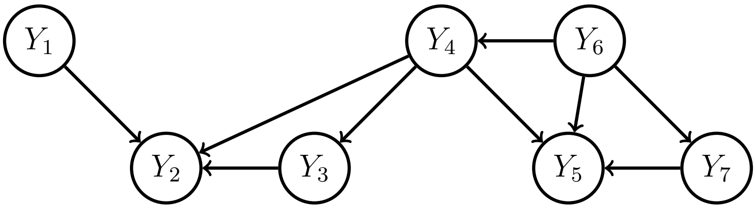

Figure 21: The output DAG for the first data set

The next interesting comparison to make is between the fitted MAG models from the greedy algorithm, and the fitted true MAG model from the MAG in Figure 18. Once again, it does not make sense to check matching edges, as the algorithm is only accurate up to equivalence class [15]. For this comparison we use the Kullback-Leibler divergence (KL-divergence) which for two continuous distributions is defined as follows.

$$KL(p|q) = \int_{-\infty}^{\infty} p(x) \log(\frac{p(x)}{q(x)}) dx$$

Our models have a resulting multivariate normal distribution, and for two multivariate distributions over a vector $x \in \mathbb{R}^k$ such as:

$$p(x) = N(x; \mu_1, \Sigma_1)$$

$$q(x) = N(x; \mu_2, \Sigma_2)$$

Following some algebra, we have that the KL-divergence has the following form[17]:

$$KL(p|q) = \frac{1}{2}[(\mu_2 - \mu_1)^T \Sigma_2^{-1} (\mu_2 - \mu_1) + tr(\Sigma_2^{-1}\Sigma_1) - \log(\frac{\det \Sigma_1}{\det \Sigma_2}) - k].$$

The KL-divergence is an example of a *pseudo-metric* [18], so we will be looking for lower values to indicate the distribution resulting from the fitted models is closer to the distribution corresponding to the true fitted MAG model. In Figure 22, we can see the KL-divergence between the distributions of the fitted MAG and DAG models and the derived MAG model from Algorithm 3.1.. The learned MAG models naturally tend to be closer in distribution to the theoretically derived MAG model, although there are two data sets which resulted in DAG models that were closer to the derived model. However, looking at the BIC scores for those samples, we see that the MAG model outperformed the respective DAG model in both cases, so this result is just a feature of those particular samples.

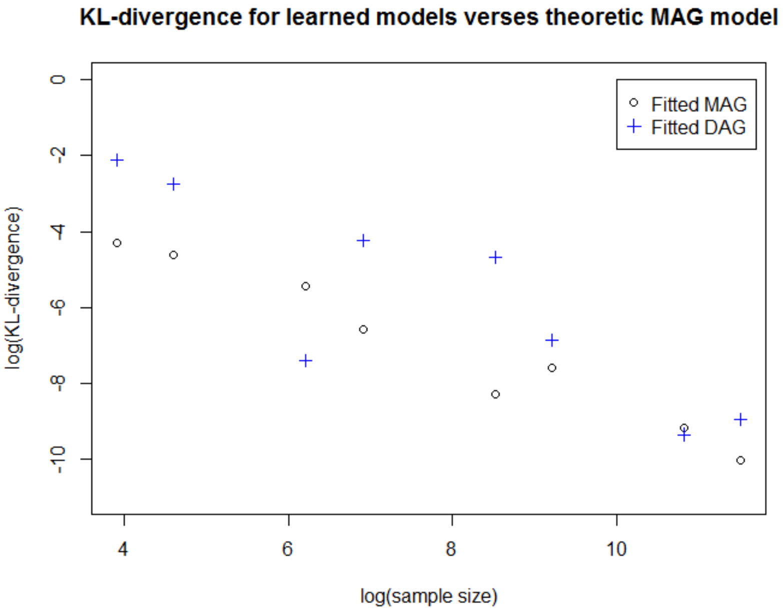

Figure 22: The KL-divergence for the distributions of the learned models verses the distribution corresponding to the theoretical MAG model

4.3 Fitting a Hidden Markov Model

A *hidden Markov model* (HMM) is a statistical Markov model where we do not observe the underlying state, as in a Markov chain, but instead a sequence of observations generated by a distribution dependent on the state space. This is the second type of data structure we will fit the MAG and DAG models to.

HMMs can be found in many real-world settings such as analysis of biological sequences, including DNA, and computational finance, where the underlying data sequence is often not immediately observable, but other data dependent on the sequence are available.

We will consider the discrete-time case, and so can model the HMM as a DAG model with the graph found in figure 23. In that graph, the X_i variable represents the Markov process at time i, and the Y_i represents the observation generated at time i.

This is distinct from the DAG case we have already examined, as when we marginalize over the X_i variables using the algorithm for MAG marginalization we notice that all of the Y_i variables are spouses of each other, so there will be no independence implications in theory. In reality, the correlation between Y_i and Y_j where the difference between i and j is large will be very small, so we will observe edges being dropped between those variables, depending on the data.

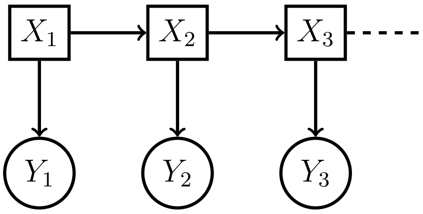

Figure 23: The beginning of a DAG representation of a hidden Markov model

We will be considering the case where the observed variables follow a univariate normal distribution, with parameters determined by the state of the underlying Markov process. The simulated data points will consist of values for the observed data for the first 10 time points, and there will be a varying number of data points in each simulation from 50 up to 100,000. There will be 8 simulations in total. Using the same greedy algorithm as before we obtained the BIC scores in Figure 24.

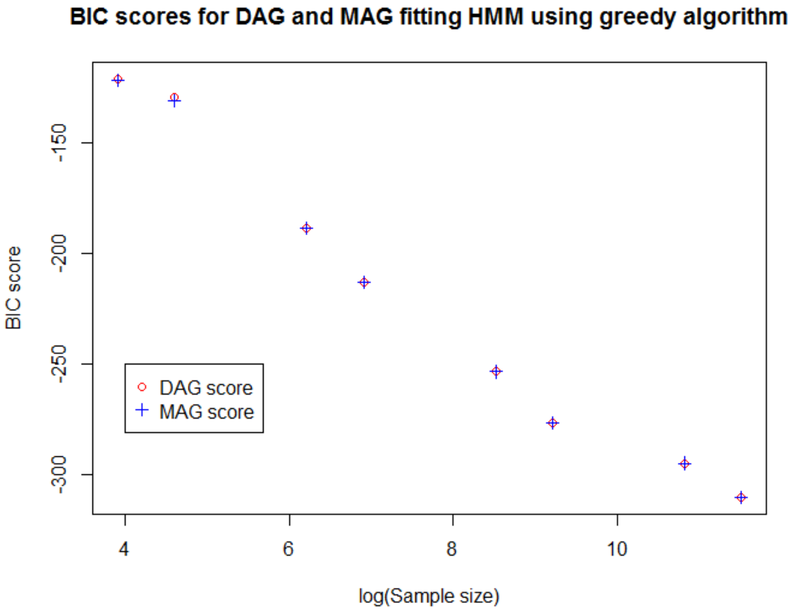

Figure 24: BIC scores for fitting to HMM observations

The conclusion that we reach is that for large enough sample sizes, the DAG and MAG models fit the HMM data very similarly. For the six largest samples, the two algorithms produced Markov equivalent graphs, so there is little benefit to using the MAG model as it takes longer to fit. We now look at the structure of the output models for large and small sample sizes. For a sample of size 50, the algorithm produced the MAG in Figure 25. It is not similar at all to the true underlying MAG, because for such a small sample the correlation between "far apart" variables such as Y_{10} and Y_2 is low.

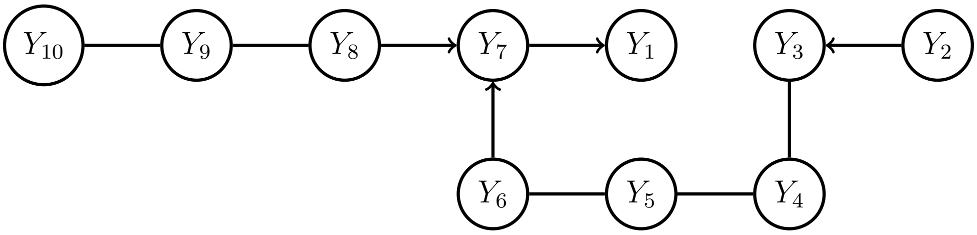

Figure 25: The output MAG for the data set of size 50

For a sample of size 100,000 the graph appears as in Figure 26. We notice that we now have, up to equivalence class, edges between vertices corresponding to observations observed two time intervals from each other.

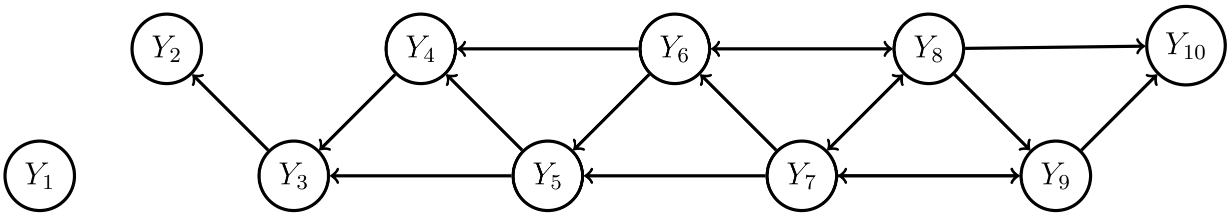

Figure 26: The output MAG for the data set of size 100,000

As we know the MAG model which captures all of the conditional independences, the one with a bidirected edges between every pair of vertices apart from Y_1, as the equivalent latent variable X_1 does not have a random starting point, it makes sense to compare our fitted models with this model to assess how they are performing.

Using KL-divergence, we compared distributions resulting from the fitted models with the distributions from the fitted theoretical MAG model and obtained the results in Figure 27. The results indicate that as expected, the distribution of the learned MAG model is closer to the distribution corresponding to the true MAG model when the sample size is larger. For large enough samples, we would expect to see every pair of variables eventually becoming correlated.

4.4 Simulation using Trees

Finally, it is useful to consider samples where the underlying graphical model is unknown. To do this we will consider a mixture of trees. We generated 10 random

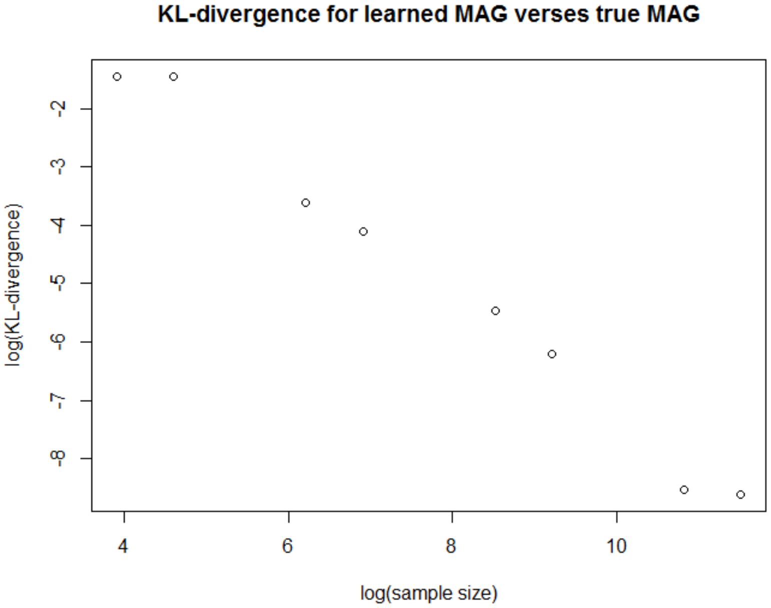

Figure 27: KL-divergence between true and learned MAG models

trees of the form in Figure 28, with the seven Y_i nodes, which are all included in our models. In order to generate samples, we select a random tree at a time, and then take one observation from that tree, which behaves just like a standard DAG model. This way, the data will not come from a single graphical model, but rather a combination of the 10. This means that any conditional independences in the data should be common to all 10 trees, so as this is unlikely we would expect all the variables to be correlated.

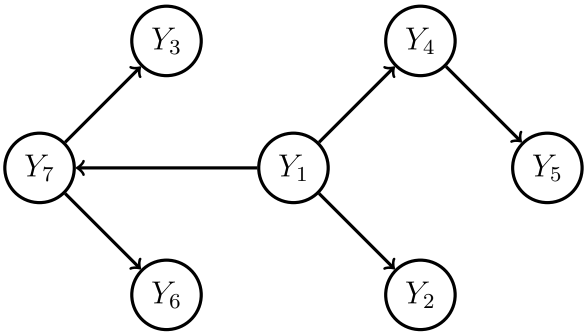

Figure 28: One example of the tree structure used to simulate our sample data

Because this kind of data does not have a single underlying graphical structure, there will be no "true" graphical model to compare the output graphs to, but rather we will be comparing between the fitted MAG and DAG models. First we consider the BIC score of the fitted DAG and MAG models, which appear in Figure 29. The algorithm produced Markov equivalent graphs when it assumed a MAG or DAG structure to the samples, for almost all data sets.

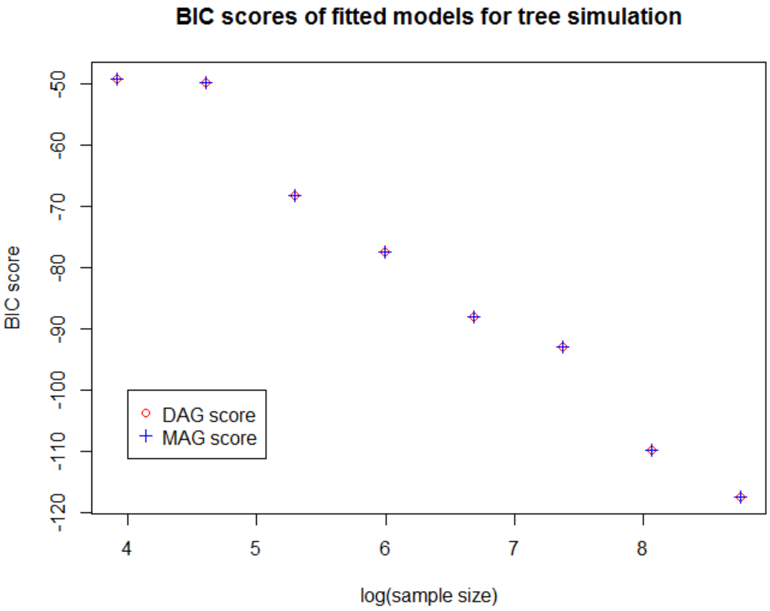

Figure 29: BIC scores for the output of the greedy algorithm on the tree simulation

4.5 Discussion

Overall, the three simulations showed that often there is not a significant difference between the MAG and DAG models. This appears to be the case when the structure behind the data is variable, for example in the trees simulation, or when the conditional independences of the observed variables can be described using a DAG model, as in the case of the hidden Markov model simulation. We learned that sample size makes a large difference to the goodness of fit. In the HMM case, increasing the sample size allowed us to observe more of the dependence between the observed variables.

Using the assumption of a MAG model to fit the data becomes particularly useful when it is known that the distribution including hidden variables can be described using a DAG model. The first simulation demonstrated the limitations of using a DAG model, as it was unable to describe the data as well as the fitted MAG model, and as the sample size increases the difference in BIC increased as well. The algorithm appears to be tending to the theoretical fitted MAG model in terms of KL-divergence, so it is no surprise that it outperforms the DAG model.

5 Conclusions

We have presented an application of the greedy algorithm[4] to learn the structure of MAG and DAG models, with a view to addressing the problem of the class of DAG models not being closed under marginalisation. The method was not too computationally taxing, and provides a simple way of learning a possible structure to a data set. The result of fitting to various simulated data sets provided mixed results, with the most successful application being to the margin of a DAG model simulation, as expected. Given that the class of MAG models contains the class of DAG models, it is not a surprise that they produce similar results in cases where the underlying structure of the variables is unknown, especially as the BIC score heavily penalises adding parameters, and undirected and bi-directed arrows of MAGs add more parameters. The sources of error in fitting encountered here were mainly down to sampling variability. By using an approach which aims to find only the equivalence class of the graphical model, we are unable to capture the causal effects without further information.

The relevance of this is that very rarely can we observe the whole of a Bayesian network, so the fact that MAG models can accommodate the independence relationships between observed variables provides a simple way to approach this problem, without having to introduce hyper-edges. There is a compromise because a MAG model does not retain all other constraints that the margin of a DAG implies.

Extensions of this might be to apply a greedy algorithm to learn the structure of marginal DAGs [2] or chain graphs, as for these more constraints implied by DAGs will hold in the marginal distribution under the assumption that it can be modelled using a chain graph or mDAG model. Validation that the MAG model will capture the marginal distribution of a DAG model against real data sets would also be useful.

References

[1] Steffen L Lauritzen. *Graphical models*, volume 17. Clarendon Press, 1996.

[2] Robin J Evans. Graphs for margins of Bayesian networks. *Scandinavian Journal of Statistics*, 2015.

[3] Thomas Richardson and Peter Spirtes. Ancestral graph Markov models. *The Annals of Statistics*, 30(4):962–1030, 2002.

[4] Christopher Nowzohour, Marloes H Maathuis, and Peter Bühlmann. Structure learning with bow-free acyclic path diagrams. *arXiv preprint arXiv:1508.01717*, 2015.

[5] Jan TA Koster. Marginalizing and conditioning in graphical models. *Bernoulli*, pages 817–840, 2002.

[6] Ricardo Silva. A mcmc approach for learning the structure of Gaussian acyclic directed mixed graphs. In *Statistical Models for Data Analysis*, pages 343–351. Springer, 2013.

[7] Thomas Richardson. Markov properties for acyclic directed mixed graphs. *Scandinavian Journal of Statistics*, 30(1):145–157, 2003.

[8] Steffen L Lauritzen and Nanny Wermuth. Graphical models for associations between variables, some of which are qualitative and some quantitative. *The annals of Statistics*, pages 31–57, 1989.

[9] David R Cox and Nanny Wermuth. Linear dependencies represented by chain graphs. *Statistical science*, pages 204–218, 1993.

[10] Steen A Andersson, David Madigan, and Michael D Perlman. Alternative Markov properties for chain graphs. *Scandinavian journal of statistics*, 28(1):33–85, 2001.

[11] Kayvan Sadeghi et al. Marginalization and conditioning for LWF chain graphs. *The Annals of Statistics*, 44(4):1792–1816, 2016.

[12] Nanny Wermuth, David R Cox, and Judea Pearl. Explanations for multivariate structures derived from univariate recursive regressions. *Ber. Stoch. Verw. Geb., Univ. Mainz*, 94, 1994.

[13] Nanny Wermuth. Probability distributions with summary graph structure. *Bernoulli*, 17:845–879, 2011.

[14] Mathias Drton. Discrete chain graph models. *Bernoulli*, pages 736–753, 2009.

[15] David Maxwell Chickering. Optimal structure identification with greedy search. *Journal of Machine Learning Research*, 3(Nov):507–554, 2002.

[16] Daniel Hsu, Adel Javanmard, and Sham M Kakade. Learning linear Bayesian networks with latent variables. *Journal of Machine Learning Research*, 2013.

[17] John Duchi. Derivations for linear algebra and optimization. *Berkeley, California*, 2007. `http://ai.stanford.edu/~jduchi/projects/general_notes.pdf`.

[18] Bharath K Sriperumbudur, Arthur Gretton, Kenji Fukumizu, Bernhard Schölkopf, and Gert RG Lanckriet. Hilbert space embeddings and metrics on probability measures. *Journal of Machine Learning Research*, 11(Apr):1517–1561, 2010.